

We claim:

1. A binary coded decimal adder circuit for adding two operands and an input carry to give a sum comprising:
 - 5 a first stage receiving said operands, grouping said operands into equal length blocks of contiguous bits and, for each said block, logically computing from said operands an intermediate sum vector, an intermediate carry vector, a propagate function and a generate function;
a second stage receiving said input carry and receiving from each said block of
10 said first stage the respective propagate function and generate function, and carry look ahead computing therefrom carries for each said block and an output carry; and
a third stage, having the same number of blocks as said first stage, each third stage block receiving a respective intermediate sum vector, and logically adjusting said
15 intermediate sum vector by pre-correction factors which depend upon a respective intermediate carry vector, at least two of the one of said second stage carries and said input carry; and
wherein outputs of said third stage logic blocks together represent said sum of operands.
- 20 2. The adder circuit of claim 1, wherein said blocks are 4-bits in length.
3. The adder circuit of claim 1, wherein said operands are of 16-bit length, and further wherein:
 - 25 said first stage has a like logic circuit for each 4-bit block;
said second stage has a single carry look ahead logic circuit; and
said third stage has a like sum correction logic circuit for each 4-bit block.
4. The adder circuit of claim 1, wherein said operands are of 64-bit length, and further wherein:
 - 30 said first stage has a like logic circuit for each 4-bit block;
said second stage has a first carry look ahead level of four carry look ahead logic circuits providing partial carries to a second carry look ahead level of a single carry look ahead circuit; and
said third stage has a like sum correction circuit for each 4-bit block.

5. The adder circuit of claim 1, wherein said pre-correction factors are calculated depending upon four cases of the logical combinations of two carries.
- 5 6. The adder circuit of claim 5, wherein said pre-correction factors include the carry from the intermediate carry vector and a bit pattern depending upon the relevant one of said four cases.
7. The adder circuit of claim 1, wherein said intermediate sum vector and said
10 intermediate carry vector are computed by three cascaded Half-Adder circuits.
8. The adder circuit of claim 1, wherein said pre-correction factors include one of four values decided on the basis of a carry from a preceding block and a carry from the current block.
- 15 9. A binary coded decimal adder circuit for adding two $4N$ -bit operands and an input carry to give a sum comprising:
a first stage receiving said operands, grouping said operands into N contiguous 4-bit length blocks and, for each said block, logically computing from said operands an
20 intermediate sum vector, an intermediate carry vector, a propagate function and a generate function;
a second stage receiving said input carry and receiving from each said block of said first stage the respective propagate function and generate function, and carry look ahead computing therefrom carries for each said block and an output carry; and
25 a third stage, having the same number of blocks as said first stage, each third stage block receiving a respective intermediate sum vector, and logically adjusting said intermediate sum vector by pre-correction factors which depend upon a respective intermediate carry vector, at least two of the one of said second stage carries and said input carry; and
30 wherein outputs of said third stage logic blocks together represent said sum of operands.
10. The adder circuit of claim 9, wherein said pre-correction factors are calculated depending upon four cases of the logical combinations of two carries.

11. The adder circuit of claim 10, wherein said pre-correction factors include the carry from the intermediate carry vector and a bit pattern depending upon the relevant one of said four cases.
- 5
12. The adder circuit of claim 10, wherein said pre-correction factors are calculated from the carry C_{N-3} of the previous 4-bits and the carry C_{N+1} of the current 4-bits.
13. The adder circuit of claim 9, wherein said intermediate sum vector and said
10 intermediate carry vector are computed by three cascaded Half-Adder circuits.
14. A method for performing binary coded decimal addition of two operands and an input carry to give a sum comprising the steps of:
grouping said operands into equal length blocks of contiguous bits;
15 for each said block:
computing from said operands an intermediate sum vector, an intermediate carry vector, a propagate function and a generate function;
computing carries, including an output carry, from said input carry and the respective propagate function and generate function; and
20 logically adjusting the respective intermediate sum vector by pre-correction factors which depend upon a respective intermediate carry vector, at least two of said carries and said input carry; and
combining said adjusted intermediate sum for all blocks to form said sum.
- 25 15. The method of claim 14, wherein said pre-correction factors are calculated depending upon four cases of the logical combinations of two carries.
16. The method of claim 15, wherein said pre-correction factors include the carry from the intermediate carry vector and a bit pattern depending upon the relevant one of
30 said four cases.
17. The method of claim 14, wherein said intermediate sum vector and said intermediate carry vector are computed by three cascaded Half-Adder circuits.

18. A computer program product, carried on a storage medium, for performing binary coded decimal addition of two operands and an input carry to give a sum including a computer program comprising:

code means for grouping said operands into equal length blocks of contiguous
5 bits;
and for each said block:
code means for computing from said operands an intermediate sum vector,
an intermediate carry vector, a propagate function and a generate function;
code means for computing carries, including an output carry, from said
10 input carry and the respective propagate function and generate function;
and
code means for logically adjusting the respective intermediate sum vector
by pre-correction factors which depend upon a respective intermediate
carry vector, at least two of said carries and said input carry; and
15 code means for combining said adjusted intermediate sum for all blocks to form
said sum.

19. A computer system for performing binary coded decimal addition of two operands and an input carry to give a sum comprising:

20 a storage unit for storing data and instructions to be performed by a processing
unit; and
a processing unit coupled to said storage unit, said processing unit programmed
to:
group said operands into equal length blocks of contiguous bits;
25 and for each said block to:
compute from said operands an intermediate sum vector, an intermediate
carry vector, a propagate function and a generate function;
compute carries, including an output carry, from said input carry and the
respective propagate function and generate function; and
30 logically adjust the respective intermediate sum vector by pre-correction
factors which depend upon a respective intermediate carry vector, at least
two of said carries and said input carry; and
combine said adjusted intermediate sum for all blocks to form said sum.